Guide de référence rapide pour Debian

Osamu Aoki <osamu@debian.org>
Coordinateur de traduction en Français : Guillaume Erbs <gerbs@free.fr>
'Auteurs' page 27

CVS, mar 29 sep 2003 22 :54 :44 UTC

Résumé

Ce Guide de référence rapide pour Debian (http://qref.sourceforge.net/) se propose de donner une introduction courte au système Debian comme **référence rapide**. C'est un extrait de Guide de référence pour Debian (http://qref.sourceforge.net/).

Copyright

Copyright © 2001–2003 by Osamu Aoki <osamu@debian.org>.

Ce document peut être utilisé selon les termes de la Licence Publique Générale de GNU version 2 ou suivante. (http://www.gnu.org/copyleft/gpl.html)

Il est permis de produire et distribuer des copies conformes de ce document à condition que la présente notice de copyright et la présente notice de permission soient préservées sur toutes les copies.

Il est permis de copier et distribuer des versions modifiées de ce document selon les conditions d'une copie conforme, à condition que le travail dérivé résultant soit entièrement distribué selon les termes d'une notice de permission identique à celle-ci.

Il est permis de copier et distribuer des traductions de ce document dans d'autres langues, selon les conditions pour versions modifiées ci-dessus, sauf que cette notice de permission peut être incluse sous forme d'une traduction approuvée par la Free Software Foundation à la place de l'Anglais original.

Table des matières

1 Préface **1**

1.1 Conventions du document 1

1.2 Fondements de la distribution Debian 1

2 Mise à jour d'une distribution vers `testing` **3**

2.1 Transition de APT vers la version de Woody 3

2.2 Préparation de la transition (« stable » vers « testing ») 3

2.3 Mise à jour du système Debian 4

2.3.1 Meilleure méthode de mise à jour avec `dselect` 5

3 Gestion des paquets Debian **7**

3.1 Introduction 7

3.1.1 Outils principaux 7

3.1.2 Outils pratiques 8

3.2 Bases de la gestion de paquets Debian 8

3.2.1 Installation de *tâches* avec `tasksel` ou `aptitude` 8

3.2.2 Configurer le système APT 9

3.2.3 `dselect` 9

3.2.4 `aptitude` 10

3.2.5 Commandes `apt-cache` et `apt-get` 10

3.2.6 Suivre une version de la distribution Debian 11

3.2.7 Mettre à niveau inférieur tous les paquets vers `stable` 12

3.2.8 Vue générale de `/etc/apt/preferences` 12

3.3 Commandes de survie Debian 13

3.3.1 Rechercher les bogues de Debian et demander de l'aide . . . 13
3.3.2 Résolution de problèmes de mise à jour avec APT . . . 14
3.3.3 Sauvetage avec `dpkg` . . . 14
3.3.4 Récupérer les données de sélection des paquets . . . 15
3.3.5 Sauver un système après avoir perdu `/var` . . . 15
3.3.6 Installer un paquet sur un système qui ne démarre plus . . . 16
3.3.7 Que faire si `dpkg` est cassé . . . 16
3.4 Commandes « nirvana » de Debian . . . 17
3.4.1 Informations sur un fichier . . . 17
3.4.2 Informations sur un paquet . . . 17
3.4.3 Installation automatique avec APT . . . 18
3.4.4 Reconfigurer les paquets installés . . . 18
3.4.5 Enlever et purger des paquets . . . 19
3.4.6 Garder de vieux paquets . . . 19
3.4.7 Système mixte `stable/ testing/ unstable` . . . 20
3.4.8 Supprimer les paquets du cache . . . 20
3.4.9 Enregistrer/copier la configuration du système . . . 20
3.4.10 Porter un paquet vers le système `stable` . . . 21
3.4.11 Archive locale de paquets . . . 22
3.4.12 Convertir ou installer un paquet non Debian . . . 22
3.4.13 Vérifier les fichiers de paquets installés . . . 23
3.5 Autres particularités de Debian . . . 23
3.5.1 La commande `dpkg-divert` . . . 23
3.5.2 Le paquet `equivs` . . . 23
3.5.3 Commandes de rechange . . . 24
3.5.4 `init` System-V et niveaux de fonctionnement . . . 24
3.5.5 Services désactivés . . . 25

A Annexe **27**

A.1 Auteurs . . . 27
A.2 Garanties . . . 28
A.3 Retour . . . 29

Chapitre 1

Préface

Mon principe est de **le garder court et simple**.

1.1 Conventions du document

Le « Guide de référence rapide pour Debian » procure de l'information par le biais de commandes simples en shell BASH.

Références :
- une page de **manuel Unix** est donnée dans la forme `bash(1)`.
- une page **GNU TEXINFO** est donnée dans la forme `info libc`.

1.2 Fondements de la distribution Debian

Debian se présente sous la forme simultanée de 3 éditions :

- stable : Convient pour un serveur de production. Ennuyeux pour une station de travail (ST).
- testing : Convient pour une ST.
- unstable : Ne pas charger cette édition aveuglément.

Lisez au moins la liste de discussion centrale `debian-devel-announce@lists.debian.org` pour des mises à jour sur l'état de Debian. (NdT : Pour les francophones, vous trouverez de l'aide en vous abonnant à la liste de discussion francophone `debian-user-french-request@lists.debian.org`.)

Au mois de mars 2002, ces trois versions correspondent à `Potato` (qualité de production), `Woody` (en beta-test, très stable maintenant), et `Sid` (en alpha-test). En août 2002, juste après la sortie de `Woody`, cela correspond à `Woody` (qualité de production), `Sarge` (en beta-test, elle sera assez difficile pendant quelque temps), et `Sid` (toujours en alpha-test). Lorsque les paquets dans `unstable` n'ont plus de bogues critiques (Release Critical, RC) répertoriés pendant environ une semaine de test, ils sont automatiquement inclus dans `testing`.

En théorie, il y a deux choses que vous pouvez faire pour disposer des dernières versions des applications.

- 'Mise à jour d'une distribution vers `testing`' page suivante (principalement pour des stations de travail)
- 'Porter un paquet vers le système `stable`' page 21 (principalement pour des serveurs)

Chapitre 2

Mise à jour d'une distribution vers `testing`

2.1 Transition de APT vers la version de Woody

Une mise à jour par le réseau du système APT et de certains paquets principaux vers la version de Woody peut être effecttuée comme suit après avoir inclus une source `stable` dans `/etc/apt/sources.list` si vous utilisez toujours Potato.

```
# apt-get update
# apt-get install libc6 perl libdb2 debconf
# apt-get install apt apt-utils dselect dpkg
```

2.2 Préparation de la transition (« stable » vers « testing »)

Suivre la version `testing` de Debian a l'effet de bord de recevoir très lentement les correctifs de sécurité. Vous êtes prévenus.

Une mise à jour via le réseau vers « testing » peut être effectuée comme suit (lancer le script suivant go-woody (`examples/`) pour le faire en une seule commande) :

Videz le fichier `/etc/apt/sources.list` existant

```
# cd /etc/apt
# cp -f sources.list sources.old
# :>sources.list
```

Récupérez une liste propre de dépôts, pour « stable »

```
# cd /
# apt-setup noprobe
 ... sélectionner http ou ftp
```

Ajoutez la section « testing » à cette nouvelle liste. Les lignes deb-src sont commentées.

```
# cd /etc/apt
# grep -e "^deb " sources.list             >sources.deb
# grep -e "^deb-" sources.list             >sources.src
# sed -e "s/stable/testing/"  sources.deb \
     >>sources.list
# sed -e "s/stable/testing/" sources.src | \
     sed -e "s/^deb-/#deb-/"  >>sources.list

# apt-get update
# apt-get install apt apt-utils
# cat >preferences <<EOF
 Package: *
 Pin: release a=testing
 Pin-Priority: 600

 Package: *
 Pin: release a=unstable
 Pin-Priority: 50

 EOF
```

Eventuellement, ajoutez la section « unstable » des archives.

```
# sed -e "s/stable/unstable/" sources.deb \
  >>sources.list
# sed -e "s/stable/unstable/" sources.src | \
 sed -e "s/^deb-/#deb-/"  >>sources.list
```

Voir 'Bases de la gestion de paquets Debian' page 8 pour configurer /etc/apt/sources.list et /etc/apt/preferences.

Vous pouvez maintenant mettre à jour, en utilisant une des méthodes décrites dans la section suivante.

2.3 Mise à jour du système Debian

Après avoir configuré correctement /etc/apt/sources.list et /etc/apt/preferences, le système peut être mis à jour vers testing. Voir 'Gestion des paquets Debian' page 7 pour les bases, et 'Résolution de problèmes de mise à jour avec APT' page 14 si vous rencontrez des problèmes.

2.3.1 Meilleure méthode de mise à jour avec `dselect`

Si le système comprend beaucoup de paquets `-dev`, etc., la méthode suivante avec `dselect` est recommandée pour un contrôle fin des paquets.

```
# dselect update # à faire avant chaque mise à jour
# dselect select # sélectionne des paquets additionnels
```

Tous vos paquets actuels seront sélectionnés lorsque `dselect` démarre. `dselect` peut vous proposer d'autres paquets selon les priorités `depends`, `suggests` et `recommends`. Si vous ne voulez pas ajouter de paquet, tapez simplement `Q` pour sortir de `dselect`.

```
# dselect install
```

Vous devrez répondre à quelques questions de configuration des paquets pendant cette partie du processus, aussi ayez vos notes à portée de main et comptez un peu de temps pour cette partie. Voir '`dselect`' page 9.

Utilisez `dselect`. **Il fonctionne toujours :)** Si vous devez mettre à jour sans `dselect` après Woody, considérez l'utilisation d'`aptitude` ou d'autres options.

Chapitre 3

Gestion des paquets Debian

Pour réduire la charge réseau des dépôts Debian, configurez un proxy HTTP local en utilisant `squid` pour les paquets téléchargés par APT et, si nécessaire, positionnez la variable d'environnement `http_proxy` ou la variable `http` dans `/etc/apt/apt.conf`. Cela permet d'améliorer considérablement les performances de la mise à jour réseau, surtout avec plusieurs machines Debian sur un LAN.

Bien que la fonction de *pinning* de `apt_preferences(5)` soit puissante, cela ne résoud pas tous les problèmes de dépendances, car les définitions de dépendances ont tendance à chercher de nouvelles version d'autres programmes fondamentaux.

L'utilisation de chroot est utile pour simultanément assurer à la fois la stabilité du système et l'accès aux dernières versions des programmes.

Ce chapitre est basé sur un système Woody, mais la plupart des informations s'appliquent aussi à un système Potato (sauf `apt_preferences(5)` et les sujets relatifs à `/etc/apt/preferences`).

3.1 Introduction

Si lire toute la documentation pour les développeurs est trop pour vous, lisez d'abord ce chapitre et commencez à goûter à la puissance de Debian avec `testing/unstable` :-)

3.1.1 Outils principaux

```
dselect   - outil de gestion des paquets à l'aide de menus (haut niveau)
dpkg      - installation de paquets (focalisé sur les fichiers de
                paquets)
apt-get   - installation de paquets (focalisé sur les archives de
                paquets, interface en ligne de commande pour APT)
tasksel   - installation de tâches (ensemble de paquets)
```

```
aptitude  - installation de paquets (paquets et tâches, interface
                ncurses pour APT)
deity     - autre interface ncurses pour APT
synaptic, gsynaptic - autres interfaces graphiques pour APT
```

Ces outils ne sont pas de même niveau. `dselect` tourne au-dessus d'APT (en ligne de commande : `apt-get`) et `dpkg`.

APT utilise `/var/lib/apt/lists/*` pour suivre les paquets disponibles tandis que `dpkg` utilise `/var/lib/dpkg/available`. Si vous avez installé des paquets directement en utilisant `apt-get` ou un programme similaire tel que `aptitude`, assurez-vous de mettre à jour le fichier `/var/lib/dpkg/available` par le menu `[U]pdate` de `dselect` ou avec la commande shell `dselect update` avant d'exécuter `dselect select`, `tasksel` ou `dpkg -l`.

Quant aux dépendances de paquets, `apt-get` récupère automatiquement les paquets marqués **depends** mais ne s'occupe pas des paquets marqués **recommends** et **suggests**, alors que `dselect` offre un contrôle avancé par menus sur le choix de ces paquets, et offre le choix des paquets en fonction des priorités **depends**, **recommends** et **suggests**. `aptitude` offre une option pour récupérer automatiquement les paquets marqués **depends**, **recommends** et **suggests**.

3.1.2 Outils pratiques

```
apt-cache         - recherche un paquet dans le cache local
dpkg-reconfigure  - reconfigure un paquet déjà installé (s'il utilise debconf
dpkg-source       - gère les paquets sources
dpkg-buildpackage - automatise la création d'un paquet
...
```

3.2 Bases de la gestion de paquets Debian

Vous pouvez installer des ensemble de paquets appelés *tâches* ou installer des paquets individuels ou mettre à jour votre système en utilisant des outils de gestion de paquets comme ci-dessous.

3.2.1 Installation de *tâches* avec `tasksel` ou `aptitude`

`tasksel` est l'**Installateur de Tâches Debian**, qui est offert comme option `simple` pendant l'installation du système.

Si on souhaite installer une fonction courante qui requiert plusieurs paquets, c'est le meilleur moyen. Assurez-vous d'exécuter les commandes dans l'ordre suivant :

```
# dselect update
# tasksel
```

`aptitude` offre aussi accès aux *tâches*. Cela vous permet de sélectionner des *tâches* mais aussi de désélectionner des paquets de la *tâche* par le menu.

3.2.2 Configurer le système APT

Pour une mise à jour sélective en fonctionnant avec la distribution `testing`, le système APT (>Woody) doit être configuré comme dans 'Transition de APT vers la version de Woody' page 3 pour utiliser les possibilités de `apt_preferences(5)`.

D'abord, ajoutez les sources pour `stable`, `testing` et `unstable` à votre fichier `/etc/apt/sources.list`. Ensuite, éditez `/etc/apt/preferences` pour configurer la bonne `Pin-Priority`.

```
Package: *
Pin: release a=stable
Pin-Priority: 500

Package: *
Pin: release a=testing
Pin-Priority: 600

Package: *
Pin: release a=unstable
Pin-Priority: 50
```

3.2.3 `dselect`

Quand `dselect` est lancé, il sélectionne automatiquement tous les paquets « Required », « Important » et « Standard ». Sur le système Potato, certains programmes, comme teTeX et Emacs, appartenaient à cette liste et il valait mieux ne pas les sélectionner lors de l'installation initiale en les désélectionnant manuellement (en tapant « _ »). Dans Woody, ils ont été déplacés dans la catégorie « Optional ».

`dselect` a une interface utilisateur déroutante. Quatre commandes importantes (en majuscules !) :

```
Touche      Action
Q           Quitter. Confirme la sélection courante et quitte
            (passe outre les dépendances).
R           Rétablir. Je ne voulais pas faire ça.
D           (Damn it!) Je me fiche de ce que dselect pense. Fais comme ça~!
U           Positionne tout dans l'état sUggéré.
```

Avec `D` et `Q`, vous pouvez sélectionner des paquets en conflit à vos risques et périls. Faites attention en utilisant ces commandes. Aujourd'hui, `dselect` est un outil de sélection par menus mûr qui offre un contrôle précis sur les choix de paquets **suggests** et **recommends**.

Ajoutez une ligne « expert » dans `/etc/dpkg/dselect.cfg` pour réduire la quantité d'informations affichée. Pour une machine plus lente, vous pouvez lancer `dselect` sur une machine plus rapide pour trouver les paquets et utiliser `apt-get install` pour les installer.

`dselect` n'offre pas accès aux paquets qui ne sont pas préférés par la `Pin-Priority`.

3.2.4 `aptitude`

`aptitude` est un nouvel installateur de paquets par menus, similaire à `dselect`. Il peut aussi être utilisé comme alternative à `dselect` en ligne de commande. Voir `aptitude(1)`.

`aptitude` accepte des commandes à une touche, généralement en minuscule.

```
Touche   Action
F10      Menu
?        Aide
u        Mise à jour des informations de l'archive de paquets
g        Téléchargement et installation des paquets sélectionnés
q        Sortie de l'écran courant et sauvegarde des changements
x        Sortie de l'écran courant sans sauvegarde
Enter    Visualisation d'information sur un paquet
```

`aptitude` offre une option pour récupérer tous les paquets automatiquement en se basant sur les priorités **depends**, **recommends** et **suggests**. Vous pouvez changer ce comportement en choisissant `F10 -> Options -> Dependency handling` dans le menu.

`aptitude` offre accès à toutes les versions des paquets.

3.2.5 Commandes `apt-cache` et `apt-get`

Pour fonctionner avec `testing` comme décrit dans l'exemple précédent, le système peut être géré avec les commandes suivantes :

- `apt-get -u upgrade`
  Cela suit la distribution `testing` et met à jour tous les paquets du système en installant leurs dépendances de `testing`.
- `apt-get -u dist-upgrade`
  Cela suit la distribution `testing` et met à jour tous les paquets du système en résolvant leurs dépendances de `testing`.
- `apt-get -u dselect-upgrade`
  Cela suit la distribution `testing` et met à jour tous les paquets du système avec la sélection de `dselect`.
- `apt-get -u install` *`package`*
  Cela installe *package* et ses dépendances de `testing`.
- `apt-get -u install` *`package`*`/unstable`
  Cela installe *package* de `unstable` en prenant les dépendances dans `testing`.

– `apt-get -u install -t unstable` *`package`*
  Cela installe *package* de `unstable` en prenant les dépendances aussi dans `unstable` en mettant `Pin-Priority` de `unstable` à 990.
– `apt-cache policy` *`foo bar ...`*
  Cela affiche l'état des paquets *foo bar . . .* .
– `apt-cache show` *`foo bar ...`* `| less`
  Cela affiche l'information sur les paquets *foo bar . . .* .
– `apt-get install` *`foo=2.2.4-1`*
  Cela installe la version *2.2.4-1* du paquet *foo*.
– `apt-get -u install` *`foo bar-`*
  Cela installe la paquet *foo* et supprime le paquet *bar*.
– `apt-get remove` *`bar`*
  Cela supprime le paquet *bar* mais garde ses fichiers de configuration.
– `apt-get remove --purge` *`bar`*
  Cela supprime le paquet *bar* et tous ses fichiers de configuration.

Dans les exemples précédents, l'utilisation de l'option `-u` permet qu'`apt-get` affiche la liste des paquets qui seront mis à jour et demande à l'utilisateur de confirmer avant de continuer. Ce qui suit permet de configurer l'option `-u` par défaut :

```
$ cat >> /etc/apt/apt.conf << .
// Always show packages to be upgraded (-u)
APT::Get::Show-Upgraded "true";
.
```

Utilisez l'option `-s` pour simuler la mise à jour sans l'effectuer.

3.2.6 Suivre une version de la distribution Debian

Selon vos préférences sur la version de Debian à suivre, vous pouvez changer le fichier `/etc/apt/preferences` d'exemple dans 'Configurer le système APT' page 9 pour refléter vos besoins :

```
suivre stable~:            changer Pin-Priority de testing à 50
suivre testing~:           garder la configuration présentée
suivre testing(unstable)~: changer Pin-Priority de unstable à 500
suivre unstable(testing)~: changer Pin-Priority de unstable à 800
```

Une idée pour le choix de Pin-Priority est de descendre la table ci-dessus au fur et à mesure que le temps après une sortie de la distribution stable avance vers le moement de geler la sortie suivante.

Attention : suivre la version `testing` de Debian a l'effet de bord de recevoir très lentement les correctifs de sécurité.

Si vous mélangez les versions de Debian, `testing` dans `stable` ou `unstable` dans `stable`, vous pourriez récupérer sans le vouloir des paquets importants de `testing` ou `unstable` qui peuvent être bogués. Vous êtes prévenus.

Des exemples de `/etc/apt/preferences` qui bloquent certains paquets importants dans leur version la plus stable en suivant la version moins stable d'autres paquets non essentiels sont disponibles dans le sous-répertoire d'exemples (`examples/`) dans les fichiers `preferences.testing` et `preferences.unstable`. Aussi, le fichier `preferences.stable` force tous les paquets à faire une mise à niveau inférieur vers `stable`.

3.2.7 Mettre à niveau inférieur tous les paquets vers `stable`

Pour faire une mise à niveau inférieur vers `stable`, éditez `/etc/apt/preferences` comme suit :

```
Package: *
Pin: release a=stable
Pin-Priority: 1001
```

et lancez `apt-get upgrade`, ce qui force la mise à niveau inférieur grâce à Pin-priority > 1000. Attention, cela peut entraîner de petites erreurs au niveau des dépendances.

3.2.8 Vue générale de `/etc/apt/preferences`

Dans le fichier `/etc/apt/preferences` d'exemple présenté dans 'Configurer le système APT' page 9, la première ligne « Package : * » signifie que cette section s'applique à tous les paquets. Vous pouvez utiliser un nom de paquet spécifique à la place de « * » pour spécifier la `Pin-Priority` de ce paquet.

La deuxième ligne « Pin : release a=stable » signifie que `apt-get` récupère le fichier `Packages.gz` de l'archive contenant le fichier `Release` contenant la ligne « Archive : stable ».

La dernière ligne « Pin-Priority : 500 » spécifie une `Pin-Priority` de 500.

S'il existe plusieurs paquets de même nom, normalement celui ayant la plus grande `Pin-Priority` est intallé.

Les significations de `Pin-Priority` sont :

- 1001 et plus : priorités de mise à niveau inférieur
  La mise à niveau inférieur des paquets est permise.
- 100 à 1000 : priorités standard
  La mise à niveau inférieur des paquets n'est pas permise. Quelques valeurs importantes :
  - 990 : priorité donnée par les options `--target-release` ou `-t` de `apt-get(8)`.
  - 500 : priorité de tous les paquets par défaut.
  - 100 : priorité de tous les paquets installés.

– 0 à 99 : priorités non automatiques (elles sont seulement utilisées si le paquet n'est pas installé et qu'il n'existe pas d'autre version disponible)
– moins que 0 : la version n'est jamais sélectionnée

Le même effet que `--target-release` peut être obtenu en configurant `/etc/apt/apt.conf` :

```
# echo 'APT::Default-Release "testing";' >> /etc/apt/apt.conf
```

L'option de ligne de commande `--target-release` et l'option dans `/etc/apt/apt.conf` ont priorité sur l'option dans `/etc/apt/preferences`. Faites attention avec ces interférences quand vous jouez avec `/etc/apt/preferences`.

3.3 Commandes de survie Debian

Avec ces connaissances, on peut vivre une vie d'**éternelle** « mise à jour » :-)

3.3.1 Rechercher les bogues de Debian et demander de l'aide

Si vous avez des problèmes avec un paquet, consultez ces sites en premier avant de demander de l'aide ou de remplir un rapport de bogue. (`lynx`, `links` et `w3m` fonctionnent bien) :

```
$ lynx http://bugs.debian.org/
$ lynx http://bugs.debian.org/package-name  # si vous connaissez le nom du pa
$ lynx http://bugs.debian.org/bugnumber     # si vous connaissez le numéro du
```

Utilisez le moteur de recherche Google (www.google.com) en incluant « site :debian.org » comme mot-clé.

En cas de doute, lisez la documentation. Positionnez `CDPATH` comme suit :

```
export CDPATH=.:/usr/local:/usr/share/doc
```

et tapez

```
$ cd packagename
$ pager README.Debian # s'il existe
$ mc
```

3.3.2 Résolution de problèmes de mise à jour avec APT

Des problèmes de dépendances peuvent arriver lors de la mise à jour avec `unstable/testing`, comme décrit dans 'Mise à jour du système Debian' page 4. La plupart du temps, c'est parce qu'un paquet qui doit être mis à jour possède une dépendance qui n'est pas trouvée. Ces problèmes sont résolus en utilisant

```
# apt-get dist-upgrade
```

Si cela ne marche pas, répétez les opérations suivantes jusqu'à ce que le problème se résolve tout seul :

```
# apt-get upgrade -f         # continue «~upgrade~» malgré les erreurs
... ou
# apt-get dist-upgrade -f    # continue «~dist-upgrade~» malgré les erreurs
```

De mauvais scripts de mise à jour peuvent causer des problèmes récurrents. Il est recommandé de résoudre ce genre de situations en consultant les scripts `/var/lib/dpkg/info/`*`packagename`*`.{post,pre}{inst,rm}` du paquet en question et de lancer :

```
# dpkg --configure -a    # configure les paquets partiellement installés
```

Si un script se plaint d'un fichier de configuration manquant, cherchez dans `/etc` le fichier de configuration correspondant. Si un fichier de configuration avec une extension `.new` (ou quelquechose similaire) existe, renommez-le (`mv`) en un fichier sans suffixe.

Des problèmes de dépendances peuvent survenir en installant des paquets de `unstable/testing`. Il existe des moyens de passer outre les dépendances.

```
# apt-get install -f package # passe outre les dépendances non résolues
```

Une autre méthode pour résoudre ces problèmes est d'utiliser le paquet `equivs`. Voir 'Le paquet `equivs`' page 23. Voir `/usr/share/doc/equivs/README.Debian`.

3.3.3 Sauvetage avec `dpkg`

La récupération d'un `dselect` (APT) qui ne fonctionne plus en utilisant `dpkg` sans APT est possible sur un système complètement cassé :

```
# cd /var/cache/apt/archives
# dpkg -i libc6* libdb2* perl*
# dpkg -i apt* dpkg* debconf*
# dpkg -i *  # jusqu'à ce qu'il n'y ait plus d'erreurs
```

Si un paquet manque, récupérez-le avec :

```
# mc            # utilisez «~FTP link~» vers un serveur FTP Debian
```

Les paquets sur le serveur HTTP/FTP peuvent ne pas être localisés sous le répertoire habituel `/dist` mais sous le nouveau répertoire `/pool`.

Installez ensuite avec :

```
# dpkg -i /var/cache/apt/archives/packagefile.deb
```

Pour une dépendance non résolue, résolvez-la ou utilisez :

```
# dpkg --ignore-depends=package1,... -i packagefile.deb
# dpkg --force-depends -i packagefile.deb
# dpkg --force-depends --purge package
# dpkg --force-confmiss -i packagefile.deb # Installe fichier de configuratio
```

3.3.4 Récupérer les données de sélection des paquets

Si `/var/lib/dpkg/status` est corrompu, le système Debian perd les données de sélection des paquets et en est grandement affecté. Cherchez le vieux fichier `/var/lib/dpkg/status` dans `/var/lib/dpkg/status-old` ou `/var/backups/dpkg.status.*`.

Garder `/var/backups/` dans une partition séparée peut être une bonne idée puisque ce répertoire contient beaucoup de données système importantes.

Si le vieux fichier `/var/lib/dpkg/status` n'est pas disponible, vous pouvez quand même récupérer les données depuis des répertoires de `/usr/share/doc/`.

```
# ls /usr/share/doc | \
  grep -v [A-Z] | \
  grep -v '^texmf$' | \
  grep -v '^debian$' | \
  awk '{print $1 " install"}' | \
  dpkg --set-selections
# dselect --expert # réinstallez le système, utilisez dselect si besoin
```

3.3.5 Sauver un système après avoir perdu `/var`

Comme le répertoire `/var` contient des données mises à jour régulièrement, comme le mail, il peut se corrompre. Le placer dans une partition indépendante limite les risques. Si le pire arrive, vous pouvez avoir à reconstruire le répertoire `/var` pour sauver le système Debian.

Obtenez la structure du répertoire `/var` à partir d'un système Debian qui fonctionne, basé sur la même ou une plus ancienne version de Debian, par exemple `/var`, et placez la à la racine du système cassé. Ensuite

```
# cd /
# mv var var-old      # s'il reste des données utiles
# tar xvzf var.tar.gz # utilisez la structure de Woody
# aptitude            # ou dselect
```

Cela doit fournir un système qui fonctionne. Vous pouvez récupérer les sélections de paquets en utilisant la technique décrite à 'Récupérer les données de sélection des paquets' page précédente. ([FIXME] : Cette procédure nécessite plus de tests.)

3.3.6 Installer un paquet sur un système qui ne démarre plus

Démarrez Linux en utilisant une disquette/CD de sauvetage Debian ou une autre partition sur un système Linux multi-boot. Montez le système qui ne démarre plus sur `/target` et utilisez le mode d'installation chroot de `dpkg`.

```
# dpkg --root /target -i packagefile.deb
```

Configurez ensuite et résolvez les problèmes.

Au fait, si c'est seulement un `lilo` cassé qui empêche le démarrage, vous pouvez démarrer avec le disque de sauvetage standard Debian. A l'invite de démarrage, en considérant que la partition racine de Linux est sur `/dev/hda12` et que vous souhaitez le niveau de fonctionnement 3, entrez :

```
boot: rescue root=/dev/hda12 3
```

Vous avez maintenant démarré un système presque complètement fonctionnel avec le noyau de la disquette. (Il peut y avoir de petits problèmes dus au manque d'options du noyau ou de modules.)

3.3.7 Que faire si `dpkg` est cassé

Cela pose de sérieux problèmes pour l'installation des fichiers `.deb`. Les commandes suivantes vous guident pour récupérer cette situation. (Sur la première ligne, vous pouvez remplacer « links » par votre navigateur favori.)

```
$ links http://http.us.debian.org/debian/pool/main/d/dpkg/
  ... téléchargez le bon dpkg_version_arch.deb
$ su
password: *****
$ ar x dpkg_version_arch.deb
# mv data.tar.gz /data.tar.gz
# cd /
# tar xzfv data.tar.gz
```

Pour une machine `i386`, `http://packages.debian.org/dpkg` peut aussi être utilisé comme URL.

3.4 Commandes « nirvana » de Debian

A la **lumière** de ces commandes, on sera sauvé de l'éternelle lutte karmique contre l'enfer de la mise à jour afin d'atteindre le **nirvana** Debian. :-)

3.4.1 Informations sur un fichier

De l'information sur un fichier peut être trouvée avec :

```
$ dpkg {-S|--search} pattern # cherche «~pattern~» dans les paquets installés
$ zgrep -e pattern /local/copy/of/debian/woody/Contents-i386.gz
            # cherche «~pattern~» dans les fichiers de l'archive Debian
```

Sinon, utilisez les commandes spécialisées pour les paquets :

```
# apt-get install dlocate
                # crée un conflit avec slocate (version sécurisée de locate)
$ dlocate filename         # solution de rechange rapide pour dpkg -L et dpkg
...
# apt-get install auto-apt # outil d'installation de paquets à la demande
# auto-apt update          # crée le fichier db pour auto-apt
$ auto-apt search pattern
               # recherche exhaustive dans l'archive de paquets, installés o
```

3.4.2 Informations sur un paquet

Cherche et affiche les informations sur un paquet. Assurez-vous qu'apt pointe vers la bonne archive en éditant `/etc/apt/sources.list`. Si vous voulez voir les différences entre un paquet de `testing`/`unstable` et un paquet installé, utilisez `apt-cache policy`—plutôt sympa.

```
# apt-get   check         # met à jour le cache et vérifie les dépendances
$ apt-cache search  texte  # cherche un paquet à partir de "texte"
$ apt-cache policy  paquet # information sur la priorité d'un paquet
$ apt-cache show -a paquet # affiche la description d'un paquet
                                      dans toutes les distributions
$ apt-cache showpkg paquet # informations de debogage sur un paquet
# dpkg  --audit|-C         # cherche les paquets partiellement installés
```

```
$ dpkg {-s|--status} paquet ... # état et description d'un paquet installé
$ dpkg -l paquet ...       # état du paquet installé (1 ligne)
$ dpkg -L paquet ...       #liste les noms de fichiers installés par le paque
```

`apt-cache showsrc` n'est pas documenté à la sortie de Woody, mais fonctionne :)

Vous pouvez aussi trouver des informations sur les paquets (j'utilise `mc` pour naviguer) dans :

```
/var/lib/apt/lists/*
/var/lib/dpkg/{available|status}
```

La comparaison des fichiers suivants fournit des informations sur ce qu'il s'est exactement passé lors des dernières installations.

```
/var/lib/dpkg/status
/var/backups/dpkg.status*
```

3.4.3 Installation automatique avec APT

Pour une installation non interactive, ajoutez la ligne suivante dans `/etc/apt/apt.conf` :

```
Dpkg::Options {"--force-confold";}
```

C'est l'équivalent de lancer `apt-get -q -y packagename`. Parce que cela répond automatiquement « oui » à toutes les questions, cela peut causer des problèmes, aussi faites attention en utilisant cette astuce. Voir `apt.conf(5)` et `dpkg(1)`.

Vous pouvez configurer les paquets par la suite en suivant 'Reconfigurer les paquets installés' de la présente page.

3.4.4 Reconfigurer les paquets installés

Utilisez les commandes suivantes pour reconfigurer un paquet installé.

```
# dpkg-reconfigure --priority=medium package [...]
# dpkg-reconfigure --all   # reconfigure tous les paquets
# dpkg-reconfigure locales # génère de nouvelles locales
# dpkg-reconfigure --p=low xserver-xfree86 # reconfigure le serveur X
```

Faites cela pour `debconf` si vous avez besoin de changer le mode de dialogue de `debconf` de façon permanente.

Certains logiciels sont livrés avec des scripts de configuration.

```
apt-setup     - crée /etc/apt/sources.list
install-mbr   - installe un gestionnaire de Master Boot Record
tzconfig      - configure le fuseau horaire local
gpmconfig     - configure le gestionnaire de souris gpm
sambaconfig   - configure samba dans potato (woody utilise debconf)
eximconfig    - configure Exim (MTA)
texconfig     - configure teTeX
apacheconfig  - configure Apache (httpd)
cvsconfig     - configure CVS
sndconfig     - configure le système sonore
...
update-alternatives - configure la commande par défaut~; par exemple, vim pou
update-rc.d         - gestion des scripts de démarrage System-V
update-menus        - système de menus Debian
...
```

3.4.5 Enlever et purger des paquets

Enlever un paquet en gardant la configuration :

```
# apt-get remove package ...
# dpkg  --remove package ...
```

Enlever un paquet et sa configuration :

```
# apt-get remove --purge package ...
# dpkg    --purge        package ...
```

3.4.6 Garder de vieux paquets

Par exemple, pour garder `libc6` et `libc6-dev` lors de l'utilisation de `dselect` et `apt-get -u upgrade` *`paquet`*, on peut utiliser les commandes suivantes :

```
# echo -e "libc6 hold\nlibc6-dev hold" | dpkg --set-selections
```

`apt-get -u upgrade` *`paquet`* ne sera pas gêné par cela. Pour garder un paquet en forçant une mise à niveau inférieur automatique avec `apt-get -u upgrade` *`paquet`* ou `apt-get -u dist-upgrade`, ajoutez ce qui suit dans `/etc/apt/preferences` :

```
Package: libc6
Pin: release a=stable
Pin-Priority: 2000
```

Ici, l'entrée « `Package:` » ne peut pas utiliser d'entrées comme « `libc6*` ». Si vous voulez garder tous les paquets binaires liés au paquet source `glibc` synchronisés, il faut les lister explicitement.

La commande suivante liste les paquets à garder :

```
dpkg --get-selections "*"|grep -e "hold$"
```

3.4.7 Système mixte `stable`/`testing`/`unstable`

`apt-show-versions` peut lister les versions des paquets disponibles dans chaque distribution.

```
$ apt-show-versions | fgrep /testing | wc
... nombre de paquets de testing
$ apt-show-versions -u
... liste de paquets qu'il est possible de mettre à jour
$ apt-get install 'apt-show-versions -u -b | fgrep /unstable'
... met à jour tous les paquets de unstable à leur dernière version
```

3.4.8 Supprimer les paquets du cache

Les installations de paquets avec APT laissent des fichiers de paquets dans le cache `/var/cache/apt/archives` ; ce cache doit être nettoyé.

```
# apt-get autoclean # efface les fichiers de paquets inutiles
# apt-get clean     # efface tous les fichiers de paquets du cache
```

3.4.9 Enregistrer/copier la configuration du système

Pour faire une copie locale de l'état de la sélection des paquets :

```
$ dpkg --get-selections "*" >myselections   # ou utiliser \*
```

`"*"` inclut aussi les noms des paquets à purger dans *`myselections`*.

Vous pouvez transférer ce fichier vers un autre ordinateur, et l'installer avec :

```
# dselect update
# dpkg --set-selections <myselections
# apt-get -u dselect-upgrade    # ou dselect install
```

3.4.10 Porter un paquet vers le système `stable`

Pour des mises à jour partielles du système `stable`, recréer un paquet pour l'environnement utilisé à partir des sources est recommandé. Cela évite de nombreuses mises à jour de paquets à cause des dépendances. D'abord, ajouter l'entrée suivante dans `/etc/apt/sources.list` :

```
deb-src http://http.us.debian.org/debian testing \
 main contrib non-free
deb-src http://non-us.debian.org/debian-non-US testing/non-US \
 main contrib non-free
deb-src http://http.us.debian.org/debian unstable \
 main contrib non-free
deb-src http://non-us.debian.org/debian-non-US unstable/non-US \
 main contrib non-free
```

Chaque entrée de `deb-src` est ici scindée en deux lignes à cause de contraintes d'impression, mais les entrées réelles dans `sources.list` doivent tenir en une seule ligne.

Ensuite, récupérez les sources et construisez un paquet :

```
$ apt-get update  # met à jour la liste des paquets source
$ apt-get source package
$ dpkg-source -x package.dsc
$ cd package-version
  ... cherche parmi les paquets requis (Build-depends dans le fichier .dsc) e
      les installe. Il faut le paquet «~fakeroot~» aussi.

$ dpkg-buildpackage -rfakeroot

  ...ou (si pas de signature)
$ dpkg-buildpackage -rfakeroot -us -uc # utilisez «~debsign~» ensuite si néce

  ...ensuite pour installer
$ su -c "dpkg -i packagefile.deb"
```

Normalement, il n'y a besoin d'installer que peu de paquets avec un suffixe « -dev » pour satisfaire les dépendances. `debsign` est dans le paquet `devscripts`. `auto-apt` peut aider à résoudre les dépendances. L'utilisation de `fakeroot` évite une utilisation non nécessaire du compte root.

Dans Woody, ces problèmes de dépendances peuvent être simplifiés. Par exemple, pour compiler le paquet source de `pine` :

```
# apt-get build-dep pine
# apt-get source -b pine
```

3.4.11 Archive locale de paquets

Afin de créer une archive locale de paquets qui soit compatible avec les systèmes APT et dselect, un fichier `Packages` doit être créé et les fichiers de paquets ont besoin d'être archivés dans un arbre de répertoires particulier.

Un dépôt local similaire à une archive officielle Debian peut être créé avec :

```
# apt-get install dpkg-dev
# cd /usr/local
# install -d pool # les paquets physiques sont placés ici
# install -d dists/unstable/main/binary-i386
# ls -1 pool | sed 's/_.*$/ priority section/' | uniq > override
# editor override # ajustez priority et section
# dpkg-scanpackages pool override /usr/local/ \
   > dists/unstable/main/binary-i386/Packages
# cat > dists/unstable/main/Release << EOF
Archive: unstable
Version: 3.0
Component: main
Origin: Local
Label: Local
Architecture: i386
EOF
# echo "deb file:/usr/local unstable main" \
   >> /etc/apt/sources.list
```

Autrement, un dépôt local peut être créé de façon rapide et sale :

```
# apt-get install dpkg-dev
# mkdir /usr/local/debian
# mv /some/where/package.deb /usr/local/debian
# dpkg-scanpackages /usr/local/debian /dev/null | \
  gzip - > /usr/local/debian/Packages.gz
#  echo "deb file:/usr/local/debian ./" >> /etc/apt/sources.list
```

Ces archives peuvent être accédées à distance en fournissant ces répertoires aux méthodes HTTP ou FTP des entrées du fichier `/etc/apt/sources.list`.

3.4.12 Convertir ou installer un paquet non Debian

`alien` permet de convertir des paquets binaires fournis aux formats `rpm` de Redhat, `slp` de Stampede, `tgz` de Slackware, et `pkg` de Solaris en paquet `deb` de Debian. Si vous voulez utiliser un paquet d'une autre distribution Linux, vous pouvez utiliser `alien` pour le convertir vers votre format de paquet préféré et l'installer. `alien` supporte aussi les paquets LSB.

3.4.13 Vérifier les fichiers de paquets installés

`debsums` permet la vérification des fichiers de paquets installés grâce à des sommes de contrôle MD5. Certains paquets n'ont pas de somme de contrôle MD5. Une solution temporaire pour les administrateurs système :

```
# cat >>/etc/apt/apt.conf.d/90debsums
DPkg::Post-Install-Pkgs {"xargs /usr/bin/debsums -sg";};
^D
```

par Joerg Wendland <joergland@debian.org> (non testé).

3.5 Autres particularités de Debian

3.5.1 La commande `dpkg-divert`

`dpkg-divert` force `dpkg` à ne pas installer un fichier à sa place par défaut, mais à une place **détournée**. Les **détournements** peuvent être utilisés dans les scripts de paquets Debian pour déplacer un fichier qui crée un conflit. Les administrateurs système peuvent utiliser ceci pour recouvrir les fichiers de configuration d'un paquet, ou lorsque des fichiers (non marqués **conffiles**) ont besoin d'être préservés par `dpkg`, lors de l'installation d'une version plus récente d'un paquet qui contient ces fichiers.

```
# dpkg-divert [--add]  filename # ajoute un détournement
# dpkg-divert --remove filename # supprime un détournement
```

Ne pas utiliser `dpkg-divert` si cela n'est pas absolument nécessaire.

3.5.2 Le paquet `equivs`

Si vous compilez un programme depuis les sources, il est mieux d'en faire un paquet Debian (`*.deb`). `equivs` est le dernier recourt.

```
Package: equivs
Priority: extra
Section: admin
Description: Passer outre les dépendances de paquets.
 C'est un paquet qui peut être utilisé pour créer des paquets Debian qui ne
 contiennent que des informations sur les dépendances.
```

3.5.3 Commandes de rechange

Pour utiliser `vim` à la place de `vi`, utilisez `update-alternatives` :

```
# update-alternatives --display vi
...
# update-alternatives --config vi
  Selection    Command
-----------------------------------------------
      1        /usr/bin/elvis-tiny
      2        /usr/bin/vim
*+    3        /usr/bin/nvi

Enter to keep the default[*], or type selection number: 2
```

Les commandes de rechange du système sont maintenues dans `/etc/alternatives` en tant que liens symboliques.

Pour avoir votre environnement de fenêtres X favori, appliquez `update-alternatives` à `/usr/bin/x-session-manager` et `/usr/bin/x-window-manager`.

`/bin/sh` est un lien direct de `/bin/bash` ou `/bin/ash`. Il est plus sûr d'utiliser `/bin/bash` pour garder la compatibilité avec de vieux scripts bash mais `/bin/ash` permet d'être plus discipliné et d'être compatible POSIX. La mise à jour vers un noyau Linux 2.4 a tendance à positionner ce lien vers `/bin/ash`.

3.5.4 `init` System-V et niveaux de fonctionnement

Le niveau de fonctionnement par défaut peut être configuré dans `/etc/inittab`.

Contrairement à d'autres distributions, Debian laisse la gestion des niveaux de fonctionnement complètement à l'administrateur. La gestion de `init` de type System-V est sensée être effectuée par les scripts `update-rc.d`.

Démarrer `/etc/init.d/`*`name`* dans les niveaux 1, 2, 3 et le stopper dans les niveaux 4, 5 avec une priorité de 20 (normal) peut se faire avec :

```
# update-rc.d name start 20 1 2 3 . stop 20 4 5 .
```

Enlever le lien symbolique alors que le script dans `init.d` existe toujours peut se faire avec :

```
# update-rc.d -f name remove
```

Pour éditer le niveau de fonctionnement, je triche. J'édite manuellement en utilisant la commande `mv` à l'invite shell de `mc` et je copie les liens avec Alt-Entrée. Par exemple :

```
# mv S99xdm K99xdm # désactive xdm (X display manager)
```

Je désactive même un daemon en insérant `exit 0` au début du script `init.d`. Après tout, ce sont des fichiers de configuration.

3.5.5 Services désactivés

La distribution Debian prend très au sérieux la sécurité du système et s'attend à ce que l'administrateur soit compétent. Ainsi, la facilité d'utilisation est quelquefois mise en second plan et certains services fournis par des daemons sont livrés avec le niveau de sécurité le plus élevé, avec le moins de services (voire aucun) disponibles dans leur état d'installation par défaut.

Exécutez `ps aux` ou consultez le contenu de `/etc/init.d/*` et `/etc/inetd.conf`, si vous avez des doutes (à propos d'Exim, DHCP, ...). De plus, consultez `/etc/hosts.deny`. La commande `pidof` est aussi utile (voir `pidof(8)`).

X11 ne permet pas les connexions TCP/IP (distantes) par défaut dans les versions récentes de Debian. La redirection X est aussi désactivée dans SSH.

Annexe A

Annexe

A.1 Auteurs

Guide de référence rapide pour Debian a été démarré par Osamu Aoki <debian@aokiconsulting.com> en tant que mémo d'installation personnel et finalement appelé « Quick Reference … ». Une grande partie du contenu venait des archives de la liste de diffusion « debian-user ». Le « Manuel d'installation Debian » et les « Notes de version Debian » ont aussi été utilisés.

Suite à une suggestion de Josip Rodin, qui est très actif dans le Projet de Documentation Debian (`http://www.debian.org/doc/ddp`) (DDP) et est le responsable actuel de la « FAQ Debian », ce document a été renommé en « Guide de référence pour Debian » et fusionné avec des chapitres de la « FAQ Debian » ayant un contenu de référence. Par la suite, l'extrait « Guide de référence rapide pour Debian » a été créé.

Ce document a été modifié, traduit et amélioré par les membres de l'équipe QREF suivants :

- Relecture et réécriture en Anglais du « Quick Reference… » original
  - Osamu Aoki `<osamu@debian.org>` (leader : tout le contenu)
- Relecture et réécriture en Anglais
  - David Sewell `<dsewell@virginia.edu>` (leader : style en)
  - Brian Nelson `<nelson@bignachos.com>`
  - Daniel Webb `<webb@robust.colorado.edu>`
  - Jan Michael C Alonzo `<jmalonzo@spaceants.net>`
- Traduction en Français
  - Guillaume Erbs `<gerbs@free.fr>` (leader : fr)
  - Rénald Casagraude `<rcasagraude@interfaces.fr>`
  - Jean-Pierre Delange `<delange@imaginet.fr>`
  - Daniel Desages `<daniel@desages.com>`
- Traduction en Italien
  - Davide Di Lazzaro `<mc0315@mclink.it>` (leader : it)
- Traduction en Portugais (Brésil)
  - Paulo Rogério Ormenese `<pormenese@uol.com.br>` (leader : pt-br)
  - Andre Luis Lopes `<andrelop@ig.com.br>`

    - Marcio Roberto Teixeira <marciotex@pop.com.br>
    - Rildo Taveira de Oliveira <to_rei@yahoo.com>
    - Raphael Bittencourt Simoes Costa <raphael-bsc@bol.com.br>
    - Gustavo Noronha Silva <kov@debian.org> (coordinator)
- Traduction en Espagnol (à peu près terminée)
    - Walter Echarri <wecharri@infovia.com.ar> (leader : es)
    - José Carreiro <ffx@urbanet.ch>
- Traduction en Allemand
    - Jens Seidel <tux-master@web.de> (leader : de)
    - Willi Dyck <wdyck@gmx.net>
    - Stefan Schröder <stefan@fkp.uni-hannover.de>
    - Agon S. Buchholz <asb@kefk.net>
- Traduction en Polonais—les membres de PDDP (http://debian.linux.org.pl) suivants :
    - Marcin Andruszkiewicz
    - Mariusz Centka <mariusz.centka@debian.linux.org.pl>
    - Bartosz Fenski <fenio@debian.linux.org.pl> (leader : pl)
    - Radoslaw Grzanka <radekg@debian.linux.org.pl>
    - Bartosz 'Xebord' Janowski
    - Jacek Lachowicz
    - Mateusz Prichacz <mateusz@debian.linux.org.pl>
    - Marcin Rogowski
    - Pawel Rozanski
    - Mariusz Strzelecki
    - Krzysztof Scierski
    - Przemyslaw Adam Smiejek <tristan@debian.linux.org.pl>
    - Mateusz Tryka <uszek@debian.linux.org.pl>
    - Cezary Uchto
    - Krzysztof Witkowski <tjup@debian.linux.org.pl>
    - Bartosz Zapalowski <zapal@debian.linux.org.pl>
- Traduction en Chinois (simplifié)
    - Hao "Lyoo" LIU <iamlyoo@163.net> (leader : zh-cn)
    - Ming Hua <minghua@rice.edu>
- Traduction en Chinois (traditionnel)
    - Tang Wei Ching <wctang@csie.nctu.edu.tw> (leader : zh-tw)

A.2 Garanties

Comme je ne suis pas un expert, je ne prétends pas tout connaître sur Debian ou Linux en général. Les considérations de sécurité que j'utilise peuvent être utilisées seulement dans le cas d'une utilisation personnelle.

Ce document ne remplace aucun guide officiel.

Toute garantie est désavouée. Toutes les marques de commerce sont la propriété de leurs propriétaires respectifs.

A.3 Retour

Les commentaires et additions à ce document sont toujours les bienvenus. Merci d'envoyer un courriel au système Debian BTS (`http://bugs.debian.org/`) sous le paquet `debian-reference` ou le paquet correspondant à la traduction. L'utilisation de `reportbug` rend facile de remplir un rapport de bogue complet. Vous pouvez toujours envoyer un courriel à Osamu Aoki (`http://people.debian.org/~osamu/`) `<osamu@debian.org>` en Anglais ou à chacun des traducteurs dans leur langue respective.